

IBM Cúram Social Program Management



# Cúram-Cache

*Version 6.0.5*



IBM Cúram Social Program Management



# Cúram-Cache

*Version 6.0.5*

**Hinweis**

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die Informationen in „Bemerkungen“ auf Seite 15 gelesen werden.

**Überarbeitung: Mai 2013**

Diese Ausgabe bezieht sich auf IBM Cúram Social Program Management v6.0.5 und alle nachfolgenden Releases, sofern nicht anderweitig in neuen Ausgaben angegeben.

Licensed Materials - Property of IBM.

© Copyright IBM Corporation 2012, 2013.

© Cúram Software Limited. 2011. Alle Rechte vorbehalten.

---

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b> . . . . .	<b>v</b>	3.7 Globale Caches in Transaktionskontext verwenden	9
<b>Tabellen</b> . . . . .	<b>vii</b>	3.8 Codebeispiele . . . . .	9
<b>Kapitel 1. Einführung</b> . . . . .	<b>1</b>	3.9 Cacheladeprogramm . . . . .	9
1.1 Zweck . . . . .	1	3.10 Cache-Client . . . . .	9
1.2 Zielgruppe . . . . .	1	3.11 Cacheinvalidierung . . . . .	10
<b>Kapitel 2. Cúram-Cache</b> . . . . .	<b>3</b>	<b>Kapitel 4. Lokale Caches für Threads</b>	<b>11</b>
2.1 Erklärung des Cúram-Cache . . . . .	3	4.1 Übersicht . . . . .	11
2.2 Konfiguration . . . . .	3	4.2 Konfiguration . . . . .	11
2.3 Statistik . . . . .	4	4.3 Codebeispiele . . . . .	11
2.4 Cúram-Cache herunterfahren. . . . .	5	<b>Kapitel 5. Lokale Caches für Transakti-</b>	
<b>Kapitel 3. Globale Caches</b> . . . . .	<b>7</b>	<b>onen.</b> . . . . .	<b>13</b>
3.1 Einführung. . . . .	7	5.1 Übersicht . . . . .	13
3.2 Anbieter für globalen Cache . . . . .	7	5.2 Konfiguration . . . . .	13
3.3 Standardgruppe für globalen Cache . . . . .	7	5.3 Codebeispiele . . . . .	13
3.4 Globale Caches . . . . .	7	<b>Bemerkungen.</b> . . . . .	<b>15</b>
3.5 Verwendungsempfehlungen . . . . .	8	Informationen zu Programmierschnittstellen . . . . .	17
3.6 Konfiguration . . . . .	8	Marken. . . . .	17



---

## Abbildungsverzeichnis

1. Cache konfigurieren . . . . . 4
2. Alle Caches in einer Gruppe konfigurieren . . . . . 4
3. Cache für einen Batchprozess inaktivieren . . . . . 4
4. Globalen Cache konfigurieren. . . . . 8
5. 'CacheLoaderAdapter' zur Implementierung des Cacheladeprogramms verwenden . . . . . 9
6. Cacheladeprogramm registrieren und Cache verwenden. . . . . 10
7. Cacheeintrag im Server-Code ungültig machen . . . . . 10
8. Lokalen Cache für Thread `curam.myproject.mycache` konfigurieren . . . . . 11
9. Lokalen Cache für einen Thread konfigurieren und verwenden . . . . . 11
10. Lokalen Cache für Transaktion `curam.myproject.mycache` konfigurieren . . . . . 13
11. Lokalen Cache für eine Transaktion konfigurieren und verwenden. . . . . 13





---

## Tabellen



---

# Kapitel 1. Einführung

---

## 1.1 Zweck

Der Zweck dieses Dokuments ist die Einführung in den Cúram-Cache, einen generischen Caching-Service, der dafür konzipiert ist, die Anforderungen an kurzfristig und langfristig aktive Caches zu erfüllen.

---

## 1.2 Zielgruppe

Dieses Handbuch ist für Architekten und Anwendungsentwickler konzipiert, die beabsichtigen, ihre Cachinganforderungen mithilfe des Cúram-Cache zu erfüllen.



---

## Kapitel 2. Cúram-Cache

---

### 2.1 Erklärung des Cúram-Cache

Der Cúram-Cache ist ein generischer Caching-Service, der dafür konzipiert ist, die Anforderungen an kurzfristig und langfristig aktive Caches zu erfüllen. Der Service ist sowohl in den Client- als auch in den Server-Containern einer Anwendungsserverumgebung (Onlineanwendung) sowie in einem herkömmlichen eigenständigen Java™-Prozess (Batchprogramme) verfügbar.

Der Cúram-Cache ermöglicht die Erstellung von drei Cachetypen:

- Global - hierbei handelt es sich um globale Caches (auf JVM-Ebene).
- Lokal für Thread - diese Caches existieren so lange wie der Thread, der ihr Eigner ist.
- Lokal für Transaktion - diese Caches existieren so lange wie die aktuelle Transaktion.

Die beiden letzten Cachetypen werden in diesem Dokument als Mehrinstanzcaches (aus mehreren Instanzen bestehende Caches) bezeichnet, weil von einem Cache mit einem bestimmten Namen jederzeit mehrere Instanzen vorhanden sein können (einer für jede aktive Transaktion bzw. für jeden aktiven Thread).

---

### 2.2 Konfiguration

Die Konfiguration aller Cachetypen im Cúram-Cache ist vollkommen deklarativ und basiert auf den von der Anwendung bereitgestellten Konfigurationsmechanismen. Die Cachekonfigurationsparameter müssen zum Eigenschaftenabschnitt APP\_CACHE hinzugefügt werden.

In der aktuellen Implementierung werden von den globalen Caches sowohl größen- als auch zeitbasierte Bereinigungsrichtlinien unterstützt, während von Mehrfachinstanzcaches nur zeitbasierte Bereinigungsregeln unterstützt werden.

Folgende Cachekonfigurationsparameter können angepasst werden:

- size - Die maximale Anzahl der Elemente im Speicher. Der Standardwert beträgt 200. Der Typ lautet INT32.
- evictionPolicy - Die Richtlinie zum Bereinigen von Elementen im Speicher, wenn die maximale Anzahl an Elementen im Speicher erreicht wird. Der Standardwert lautet LRU. Der Typ ist STRING. Folgende Werte sind gültig:
  - LRU - zuletzt verwendete (least recently used)
  - LFU - am wenigsten verwendete (least frequently used)
  - FIFO - erster rein, erster raus (first in first out)
- timeToLive - Die Anzahl der Sekunden, die ein Element im Cache verbleiben darf. Wenn ein positiver Wert ungleich null festgelegt ist, werden Elemente, die sich länger als der Wert für diesen Parameter (in Sekunden) im Cache befinden, gelöscht.
- timeToIdle - Die Anzahl der Sekunden, die ein Element im Cache nicht verwendet werden darf, bevor es gelöscht wird. Wenn ein positiver Wert ungleich null festgelegt ist, werden Elemente, die länger als der Wert für diesen Parameter (in Sekunden) nicht verwendet wurden, gelöscht.

Alle Eigenschaften der Cachekonfiguration müssen die folgende Schreibweise aufweisen:

```
curam.cache.<cachegruppenname>.<cachename>.<parameter>
```

Dabei gilt Folgendes:

- <cachegruppenname> ist der Name der Cachegruppe, zu der der Cache gehört.

- <cachename> ist der Name des Cache. Er könnte auch "\*" lauten, was zur Folge hätte, dass der Konfigurationsparameter auf alle Caches in dieser Cachegruppe angewendet werden würde.
- <parameter> kann entweder für size, evictionPolicy, timeToIdle oder timeToLive stehen.

Im folgenden Beispiel sind für den globalen Cache curam.myproject.mycache in der globalen Standard-cachegruppe curam-group eine Größe von 1000 Elementen und die Bereinigungsrichtlinie 'LRU' (Least Recently Used) konfiguriert.

```
curam.cache.curam-group.curam.myproject.mycache.size=1000
curam.cache.curam-group.curam.myproject.mycache.evictionPolicy=LRU
```

Abbildung 1. Cache konfigurieren

Im folgenden zweiten Beispiel ist für den lokalen Cache der Transaktion curam.myproject.mycache in der lokalen Cachegruppe der Transaktion transaction-group eine Leerlaufzeit von 10 Sekunden konfiguriert, während für alle anderen lokalen Caches der Transaktion ein Wert von 5 Sekunden konfiguriert ist.

```
curam.cache.transaction-group.curam.myproject.mycache
    .timeToIdle=10
curam.cache.transaction-group.*.timeToIdle=5
```

Abbildung 2. Alle Caches in einer Gruppe konfigurieren

Die Daten einer Cachekonfiguration, die im Repository für die Anwendungskonfigurationen gespeichert sind, können durch Übergeben der entsprechenden Werte als JVM-Systemeigenschaften überschrieben werden. Dies kann für Batchprozesse sinnvoll sein, wenn das Anwendungsprofil von der Onlineanwendung abweichen kann.

Am folgenden Beispiel wird veranschaulicht, wie der globale Cache curam.myproject.mycache in der lokalen Standardcachegruppe für einen Batchprozess inaktiviert wird.

```
ant -f app_batchlauncher.xml
    -Dcuram.cache.curam-group.curam.myproject.mycache.size=0
    -Dbatch.userna...
```

Abbildung 3. Cache für einen Batchprozess inaktivieren

---

## 2.3 Statistik

Alle Caches im Cúram-Cache sind für Statistikdaten instrumentiert und in die Cúram-JMX-Infrastruktur integriert. Die folgende Mindestanzahl an Statistikdaten wird von jedem Cachetyp über die MBean CuramCacheStats verfügbar gemacht:

- Cachegruppe - der Name der Cachegruppe.
- Cache - der Name des Cache.
- Ebene - der Name der Cacheebene (Hauptspeicher, Festplatte, etc).
- Größe - die Anzahl der Elemente im Cache.
- Treffer - die Anzahl der Anforderungen an den Cache, für die ein Element zurückgegeben wurde, das bereits im Cache geladen war.
- Fehler - die Anzahl der Anforderungen an den Cache, für die ein Element zurückgegeben wurde, das noch in den Cache geladen werden musste.
- Bereinigungen - die Anzahl der Bereinigungen von Elementen im Cache.
- Durchschnittliche Abrufzeit (ns) - die durchschnittlich abgelaufene Zeit (in Nanosekunden), die zum Lesen eines Elements im Cache erforderlich war. Beachten Sie, dass von manchen Anbietern nur eine Angabe in Millisekunden unterstützt wird.

Für Mehrfachinstanzchaches werden Momentaufnahmestatistiken und modular aufgebaute Statistikdaten bereitgestellt. Momentaufnahmestatistiken können für alle Instanzen verwendet werden, die zum Zeitpunkt der Abfrage aktiv sind und modular aufgebaute Statistikdaten werden aus allen Instanzen berechnet, die erstellt wurden.

---

## 2.4 Cúram-Cache herunterfahren

Der Cúram-Cache muss beim Beenden der Java Virtual Machine (JVM) ordnungsgemäß heruntergefahren werden. Vom Cúram-Cache wird zwar automatisch ein Ankerpunkt zum Herunterfahren der JVM installiert, der als letztes Mittel zum Bereinigen des Cache verwendet wird, es wird jedoch empfohlen, den Cache durch Aufrufen von `CacheManager.shutdown()` explizit herunterzufahren, sofern dies möglich ist.





---

## Kapitel 3. Globale Caches

---

### 3.1 Einführung

Globale Caches sind Caches, die im Bereich eines JVM-Prozesses oder darüber hinaus vorhanden sind. In der aktuellen Version des Cúram-Cache sind globale Caches nur im Bereich eines JVM-Prozesses vorhanden. Ein Eintrag, der in einem globalen Cache gespeichert ist, besteht über die Transaktionsgrenze hinweg fort, bis er explizit, vom Entwickler oder implizit als Ergebnis der Bereinigungsrichtlinie entfernt wird, die dem Cache zugeordnet ist.

Hierbei sollte beachtet werden, dass die Daten eines globalen Cache aufgrund seiner hohen Lebensdauer dazu neigen, kurze Zeiträume mit Inkonsistenzen aufzuweisen, wenn im Cache gespeicherte Objekte aktualisiert werden. Wenn für eine Anwendung eine Aktualisierung durchgeführt wird, die sich auf ein im Cache gespeichertes Objekt auswirkt, wird der zugeordnete Cacheeintrag asynchron ungültig gemacht. Von der Cachinginfrastruktur wird zwar gewährleistet, dass ein Cacheeintrag unter Umständen ungültig gemacht wird, es kann jedoch nicht ein bestimmter maximaler Zeitrahmen garantiert werden. Wenn entschieden werden soll, ob bestimmte Anwendungsdaten im globalen Cache zwischengespeichert werden können, ist es von großer Bedeutung, mit diesem Verhalten vertraut zu sein.

---

### 3.2 Anbieter für globalen Cache

Vom Cúram-Cache werden große Teil der globalen Cachinginfrastruktur mithilfe von Cachinglösungen von Drittanbietern implementiert, die im vorliegenden Dokument als Cachinganbieter bezeichnet werden. Standardanbieter ist 'Ehcache' eine Open-Source-Lösung mit hoher Leistung für eine dezentrale Cachinginfrastruktur.

---

### 3.3 Standardgruppe für globalen Cache

Der globale Cache wird auf der Grundlage allgemeiner Konfigurationsanforderungen in Gruppen zusammengefasst, zum Beispiel für die Replikation oder den Plattenspeicher. Alle Caches in der Anwendung müssen in der Standardcachegruppe erstellt werden. Der Name der Standardcachegruppe lautet curam-group.

In der aktuellen Implementierung ist die Standardcachegruppe nicht selbstreplizierend und unterstützt nicht Plattenüberlauf und Plattenpersistenz. Da die Selbstreplizierung inaktiviert ist, sind die Cacheoperationen nur für die JVM sichtbar, in der sich der globale Cache befindet. Damit jedoch alle Caches in der Standardcachegruppe im Cluster der Anwendungsserver konsistent bleiben, wird ein Mechanismus zur expliziten Cacheinvalidierung bereitgestellt. Die Cacheinvalidierung kann nur durch Server-Code ausgelöst werden, es werden aber Caches sowohl in Server- als auch Client-Containern für alle JVMs im Cluster der Anwendungsserver ungültig gemacht.

---

### 3.4 Globale Caches

Globale Caches werden mit einem Aufruf der Methode 'get()' der Cachegruppe erstellt. Wenn noch kein Cache vorhanden ist, wird ein Cache erstellt; die Konfigurationsdaten werden vor der Rückgabe auf den Cache angewendet, falls sie vorhanden sind. Globale Caches werden in der Regel mithilfe von Cacheladeprogrammen gefüllt, die von Cache-Clients registriert werden. Durch diese Methode wird der Cache-Client von der Verwaltung des gleichzeitigen Zugriffs auf den Cache getrennt, während der Cache geladen wird.

Vom Cúram-Cache wird zwar nicht die Verwendung von serialisierbaren Objekten in der eigenen API erzwungen, bestimmte Funktionen der Cachinginfrastruktur sind jedoch nur verfügbar, wenn die Schlüssel

oder im Cache gespeicherten Objekte serialisierbar sind. Aus diesem Grund wird empfohlen, serialisierbare Schlüssel und Werte im Cúram-Cache zu verwenden, so oft dies möglich ist.

**Verwendung nicht serialisierbarer Schlüssel:** Cacheeinträge, die nicht serialisierbare Schlüssel aufweisen, werden nur für die lokale JVM und nicht im gesamten Cluster der Anwendungsserver ungültig gemacht.

---

## 3.5 Verwendungsempfehlungen

In der folgenden Liste werden Empfehlungen zur Verwendung des globalen Cache aufgeführt:

- Stellen Sie nur unveränderliche Objekte in den Cache.
- Verwenden Sie soweit möglich serialisierbare Schlüssel und Werte. Zumindest die Schlüssel sollten serialisierbar sein.
- Verwenden Sie zum Füllen des Cache ein Cacheladeprogramm. So kann der Cache die Vorteile der Optimierungen für differenzierten gleichzeitigen Zugriff nutzen, die in den Cacheanbieter integriert sind und der Benutzer muss den gleichzeitigen Zugriff auf den Cache nicht selbst verwalten.
- Das Laden des Cache ohne ein Ladeprogramm (mithilfe der Aufrufe `get()` und `put()`) muss aus zwei Gründen vermieden werden:
  - Verwaltung des gleichzeitigen Zugriffs - In diesem Fall ist der Benutzer für die Verwaltung des gleichzeitigen Zugriffs auf den Cache verantwortlich, während der Cache geladen wird. Der Benutzer hat zwei Möglichkeiten:
    - Steuerung des gleichzeitigen Zugriffs auf die Codeblöcke `get()` und `put()` - diese Methode wird in einem leistungskritischem Bestandteil einer Anwendung nicht empfohlen, bietet dafür jedoch die Garantie, dass ein Objekt nur einmal geladen wird.
    - Zulassen des gleichzeitigen Zugriffs auf die Codeblöcke `get()` und `put()` - diese Methode eignet sich besser für eine höhere gleichzeitige Nutzung, ein Objekt kann in diesem Fall jedoch mehrfach von unterschiedlichen Threads geladen werden.
  - Effiziente Datenverwaltung - ohne ein Cacheladeprogramm muss der Cache vorab mit allen Daten gefüllt werden. Bei Verwendung eines Cacheladeprogramms werden nur die erforderlichen Daten in den Cache extrahiert.
- Verwenden Sie Cachennamen, die über ein Präfix aus einem Paketnamen verfügen, der im Projekt eindeutig ist. Beispiel: `curam.cpm.myCache` ist ein geeigneter Name für einen Cache in einem Cúram Provider Management™-Projekt.

---

## 3.6 Konfiguration

Von allen globalen Caches werden die folgenden Standardwerte für die Konfigurationsparameter übernommen.

- `size` - 200
- `evictionPolicy` - LRU
- `timeToLive` - 0 (nicht aktiv)
- `timeToIdle` - 0 (nicht aktiv)

Diese Werte können durch den globalen Cache überschrieben werden. Im folgenden Beispiel ist `curam-group` der Name der Standardcachegruppe und `curam.myproject.mycache` der Name des Cache.

```
curam.cache.curam-group.curam.myproject.mycache.size=1000
curam.cache.curam-group.curam.myproject.mycache.evictionPolicy=LRU
curam.cache.curam-group.curam.myproject.mycache.timeToLive=3600
curam.cache.curam-group.curam.myproject.mycache.timeToIdle=300
```

*Abbildung 4. Globalen Cache konfigurieren*

---

## 3.7 Globale Caches in Transaktionskontext verwenden

Wenn ein globaler Cache in einem Transaktionskontext verwendet wird, muss sorgfältig sichergestellt werden, dass die Konsistenz des Cache erhalten bleibt, wenn die aktuelle Transaktion rückgängig gemacht wird.

**Cacheinvalidierung im Transaktionskontext:** Entfernen oder aktualisieren Sie im Cache gespeicherte Elemente nicht direkt, wenn Daten geändert werden, die den Inhalt eines Cache betreffen. Rufen Sie stattdessen die Methode `CacheManagerEjb.postInvalidationMessage()` auf, um eine Invalidierungsnachricht zu senden, von der die Cacheinvalidierung ausgelöst wird.

---

## 3.8 Codebeispiele

Dieser Abschnitt enthält Codebeispiele, die veranschaulichen, wie ein Cacheladeprogramm geschrieben wird, wie der Cache verwendet wird und wie ein Cacheeintrag ungültig gemacht wird.

---

## 3.9 Cacheladeprogramm

Im folgenden Beispiel wird die Klasse 'CacheLoaderAdapter' für die Implementierung von 'MyCacheLoader' verwendet.

```
...
public class MyCacheLoader extends
    CacheLoaderAdapter<Integer, ReadWorkQueueDetails> {
    /* (non-Javadoc)
     * @see curam.util.cache.CacheLoader#load(java.lang.Object)
     */
    public ReadWorkQueueDetails load(Integer workQueueID)
        throws ApplicationException, InformationalException {
        WorkAllocation wa = (WorkAllocation)WorkAllocationFactory
            .newInstance();
        ReadWorkQueueKey key = new ReadWorkQueueKey();
        key.key = new ReadWorkQueueKey();
        key.key.key = new WorkQueueKey();
        key.key.key.workQueueID = workQueueID;
        ReadWorkQueueDetails item = wa.readWorkQueue(key);
        if(item != null) {
            return item.dtls;
        }
        return null;
    }
}
...
```

Abbildung 5. 'CacheLoaderAdapter' zur Implementierung des Cacheladeprogramms verwenden

---

## 3.10 Cache-Client

Am folgenden Beispiel wird die übliche Vorgehensweise zum Registrieren eines Cacheladeprogramms und Verwenden des Cache beschrieben.

```

...
public class MyCacheClient {
    // keep a static reference to mycache
    private static Cache<Integer,
        ReadWorkQueueDetails> myCache;

    static {
        // retrieve a reference to mycache and register
        // the cache loader
        myCache = CacheManager.getDefaultCacheGroup()
            .getCache("mycache");
        myCache.registerCacheLoader(new MyCacheLoader());
    }

    public WorkAllocation() {
...
    }
...
    // use the cache
    ReadWorkQueueDetails wq = myCache.get(1);
...

```

Abbildung 6. Cacheladeprogramm registrieren und Cache verwenden

---

### 3.11 Cacheinvalidierung

Am folgenden Beispiel wird veranschaulicht, wie ein Cacheeintrag in einem Code ungültig gemacht wird, der in einem Transaktionskontext (Server-Code) ausgeführt wird. Wie in 3.3, „Standardgruppe für globalen Cache“, auf Seite 7 beschrieben, kann die Cacheinvalidierung für globale Caches in der Standardcachegruppe nur durch Server-Code ausgelöst werden.

```

...
CacheManagerEjb.postInvalidationMessage(
    new CacheInvalidationMessage<Key>("mycache", 1));
...

```

Abbildung 7. Cacheeintrag im Server-Code ungültig machen

---

## Kapitel 4. Lokale Caches für Threads

---

### 4.1 Übersicht

Die aufgeführten Caches sind eng mit dem Thread verknüpft, der zu ihrer Erstellung verwendet wird. Von keinem anderen Thread kann auf die Daten in diesen Caches zugegriffen werden und die Caches werden nur gelöscht, wenn der Thread beendet wird, von dem sie erstellt wurden. Die lokalen Caches für Threads sind sehr spezialisiert. Sie dürfen nur für kleine Caches verwendet werden, wenn der Aufwand für die Steuerung eines Multithread-Zugriffs für den globalen Cache nicht zumutbar ist.

---

### 4.2 Konfiguration

Von den lokalen Caches für Threads wird nur eine zeitbasierte Bereinigungsrichtlinie unterstützt. Die beiden einzigen verwendbaren Konfigurationsparameter und ihre jeweiligen Standardwerte lauten wie folgt:

- `timeToLive` - 0 (nicht aktiv)
- `timeToIdle` - 0 (nicht aktiv)

Der Name der Gruppe für die lokalen Caches für Threads lautet `thread-group`. Dieser Name muss zum Konfigurieren des lokalen Cache für einen Thread wie im nachfolgenden Beispiel beschrieben verwendet werden.

```
curam.cache.thread-group.curam.myproject.mycache.timeToLive=60
curam.cache.thread-group.curam.myproject.mycache.timeToIdle=10
```

*Abbildung 8. Lokalen Cache für Thread `curam.myproject.mycache` konfigurieren*

---

### 4.3 Codebeispiele

Auf lokale Caches für Threads darf nur zugegriffen werden, wenn der korrekte Kontext (der korrekte Thread) verwendet wird. Es wird zum Beispiel nicht empfohlen, einen lokalen Cache für einen Thread in einem statischen Codeblock zu konfigurieren, da es vorkommen kann, dass dieser Thread nicht mit dem Thread übereinstimmt, der den Cache später verwendet.

```
public void myMethod() {
    ...
    Cache<String, String> threadCache = CacheManager.
        getThreadLocalCacheGroup().getCache("mycache");
    String value = threadCache.get("key");
    if(value == null) {
        // perform expensive operation to calculate value - this
        // processing only happens once for each thread
        ...
        // and store the result
        threadCache.put("key", "value");
    }
    ...
}
```

*Abbildung 9. Lokalen Cache für einen Thread konfigurieren und verwenden*



---

## Kapitel 5. Lokale Caches für Transaktionen

---

### 5.1 Übersicht

Ein lokaler Cache für eine Transaktion ist ein Cache, der nur für die Dauer der aktuellen Transaktion besteht. Dieser Cachetyp ist nur in einer Serveranwendung verfügbar.

### 5.2 Konfiguration

Von den lokalen Caches für Transaktionen wird nur eine zeitbasierte Bereinigungsrichtlinie unterstützt. Die beiden einzigen verwendbaren Konfigurationsparameter und ihre jeweiligen Standardwerte lauten wie folgt:

- `timeToLive` - 0 (nicht aktiv)
- `timeToIdle` - 5

Der Name der Gruppe für die lokalen Caches für Transaktionen lautet `transaction-group`. Dieser Name muss zum Konfigurieren von lokalen Caches für Transaktionen wie im nachfolgenden Beispiel beschrieben verwendet werden.

```
curam.cache.transaction-group.curam.myproject.mycache
    .timeToLive=60
curam.cache.transaction-group.curam.myproject.mycache
    .timeToIdle=10
```

Abbildung 10. Lokalen Cache für Transaktion `curam.myproject.mycache` konfigurieren

### 5.3 Codebeispiele

Wie auf lokale Caches für Threads darf auch auf lokale Caches für Transaktionen nur zugegriffen werden, wenn der korrekte Kontext (die korrekte Transaktion) verwendet wird.

```
public void myMethod() {
    ...
    Cache<String, String> txnCache = CacheManagerEjb.
        getTransactionLocalCacheGroup().getCache("mycache");
    String value = txnCache.get("key");
    if(value == null) {
        // perform expensive operation to calculate value - this
        // processing only happens once per transaction
        ...
        // and store the result
        txnCache.put("key", "value");
    }
    ...
}
```

Abbildung 11. Lokalen Cache für eine Transaktion konfigurieren und verwenden





---

## Bemerkungen

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden. Möglicherweise bietet IBM die in dieser Dokumentation beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim zuständigen IBM Ansprechpartner erhältlich. Hinweise auf IBM-Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte oder Services von IBM verwendet werden können. Anstelle der IBM Produkte, Programme oder Services können auch andere, ihnen äquivalente Produkte, Programme oder Services verwendet werden, solange diese keine gewerblichen oder anderen Schutzrechte von IBM verletzen. Die Verantwortung für den Betrieb von Produkten, Programmen und Services anderer Anbieter liegt beim Kunden. Für die in diesem Handbuch beschriebenen Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieses Handbuchs ist keine Lizenzierung dieser Patente verbunden. Lizenzanforderungen sind schriftlich an folgende Adresse zu richten (Anfragen an diese Adresse müssen auf Englisch formuliert werden):

IBM Director of Licensing

IBM Europe, Middle East & Africa

Tour Descartes

2, avenue Gambetta

92066 Paris La Defense

France

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing

Legal and Intellectual Property Law

IBM Japan Ltd.

19-21, Nihonbashi-Hakozakicho, Chuo-ku

Tokyo 103-8510, Japan

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden.

Die hier enthaltenen Informationen werden in regelmäßigen Zeitabständen aktualisiert und als Neuauflage veröffentlicht. IBM kann ohne weitere Mitteilung jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen. Verweise in diesen Informationen auf Websites anderer Anbieter werden lediglich als Service für den Kunden bereitgestellt und stellen keinerlei Billigung des Inhalts dieser Websites dar.

Das über diese Websites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt. Die Verwendung dieser Websites geschieht auf eigene Verantwortung.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht. Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängig voneinander erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

IBM Corporation

Dept F6, Bldg 1

294 Route 100

Somers NY 10589-3216

U.S.A.

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Bereitstellung des in diesem Dokument beschriebenen Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt auf der Basis der IBM Rahmenvereinbarung bzw. der Allgemeinen Geschäftsbedingungen von IBM, der IBM Internationalen Nutzungsbedingungen für Programmpakete oder einer äquivalenten Vereinbarung.

Alle in diesem Dokument enthaltenen Leistungsdaten stammen aus einer kontrollierten Umgebung. Die Ergebnisse, die in anderen Betriebsumgebungen erzielt werden, können daher erheblich von den hier erzielten Ergebnissen abweichen. Einige Daten stammen möglicherweise von Systemen, deren Entwicklung noch nicht abgeschlossen ist. Eine Gewährleistung, dass diese Daten auch in allgemein verfügbaren Systemen erzielt werden, kann nicht gegeben werden. Darüber hinaus wurden einige Daten unter Umständen durch Extrapolation berechnet. Die tatsächlichen Ergebnisse können davon abweichen. Benutzer dieses Dokuments sollten die entsprechenden Daten in ihrer spezifischen Umgebung prüfen.

Alle Informationen zu Produkten anderer Anbieter stammen von den Anbietern der aufgeführten Produkte, deren veröffentlichten Ankündigungen oder anderen allgemein verfügbaren Quellen.

IBM hat diese Produkte nicht getestet und kann daher keine Aussagen zu Leistung, Kompatibilität oder anderen Merkmalen machen. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

Aussagen über Pläne und Absichten von IBM unterliegen Änderungen oder können zurückgenommen werden und repräsentieren nur die Ziele von IBM.

Alle von IBM angegebenen Preise sind empfohlene Richtpreise und können jederzeit ohne weitere Mitteilung geändert werden. Händlerpreise können u. U. von den hier genannten Preisen abweichen.

Diese Veröffentlichung dient nur zu Planungszwecken. Die in dieser Veröffentlichung enthaltenen Informationen können geändert werden, bevor die beschriebenen Produkte verfügbar sind.

Diese Veröffentlichung enthält Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufs. Sie sollen nur die Funktionen des Lizenzprogramms illustrieren und können Namen von Personen, Firmen, Marken oder Produkten enthalten. Alle diese Namen sind frei erfunden; Ähnlichkeiten mit tatsächlichen Namen und Adressen sind rein zufällig.

**COPYRIGHTLIZENZ:**

Diese Veröffentlichung enthält Musteranwendungsprogramme, die in Quellsprache geschrieben sind und Programmier Techniken in verschiedenen Betriebsumgebungen veranschaulichen. Sie dürfen diese Musterprogramme kostenlos kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, zu verwenden, zu vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle für die Betriebsumgebung konform sind, für die diese Musterprogramme geschrieben werden. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet. IBM kann daher die Zuverlässigkeit, Wartungsfreundlichkeit oder Funktion dieser Programme nicht garantieren oder implizieren. Die Musterprogramme werden "WIE BESEHEN", ohne Gewährleistung jeglicher Art bereitgestellt. IBM übernimmt keine Haftung für Schäden, die durch Ihre Verwendung der Musterprogramme entstehen.

Kopien oder Teile der Musterprogramme bzw. daraus abgeleiteter Code müssen folgenden Copyrightvermerk beinhalten:

© (Name Ihres Unternehmens) (Jahr). Teile des vorliegenden Codes wurden aus Musterprogrammen der IBM Corp. abgeleitet.

© Copyright IBM Corp. \_Jahreszahl oder Jahreszahlen eingeben\_. Alle Rechte vorbehalten.

Wird dieses Buch als Softcopy (Book) angezeigt, erscheinen keine Fotografien oder Farabbildungen.

---

## Informationen zu Programmierschnittstellen

In der vorliegenden Veröffentlichung werden vorgesehene Programmierschnittstellen dokumentiert, mit deren Hilfe Kunden Programme für den Zugriff auf IBM Cúram Social Program Management-Services schreiben können.

---

## Marken

IBM, das IBM Logo und [ibm.com](http://www.ibm.com) sind Marken oder eingetragene Marken der International Business Machines Corporation. Weitere Produkt- und Servicenamen können Marken von IBM oder anderen Unternehmen sein. Eine aktuelle Liste der IBM Marken finden Sie auf der Website "Copyright and trademark information" unter <http://www.ibm.com/legal/us/en/copytrade.shtml>.

Java und alle auf Java basierenden Marken und Logos sind eingetragene Marken der Oracle Corporation und/oder ihrer verbundenen Unternehmen.

Sonstige Namen können Marken der jeweiligen Rechtsinhaber sein. Weitere Firmen-, Produkt- und Servicenamen können Marken oder Servicemarken anderer Unternehmen sein.







Gedruckt in Deutschland